

## AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings of claims in the application:

### LISTING OF CLAIMS

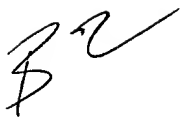
1-58 (Canceled)

- 
59. (Previously Presented) An application software program, comprising:
- an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions residing on a computer-readable medium, said instructions comprising operation codes and operands, said program operable to be loaded to and executed by a resource-constrained device, said instructions previously converted from at least one class file, said conversion transforming at least one reference to a constant pool to inline data in said instructions.
60. (Previously Presented) The software program of claim 59 wherein one or more of said references to said constant pool are transformed into inline data in operands in one or more of said instructions.
61. (Previously Presented) The software program of claim 59 wherein one or more of said references to said constant pool are transformed into inline data in operation codes in one or more of said instructions.

62. (Previously Presented) The software program of claim 59 wherein said resource-constrained device is based on a 16-bit processor architecture.
63. (Previously Presented) The software program of claim 59 wherein said resource-constrained device is based on an 8-bit processor architecture.
64. (Previously Presented) The software program of claim 59 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 64 kilo-bytes.
- B<sup>2</sup> 65. (Previously Presented) The software program of claim 59 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 4 kilo-bytes.
66. (Previously Presented) The software program of claim 59 wherein said instructions are configured for execution by a virtual machine running on a microprocessor residing on said resource-constrained device.
67. (Previously Presented) The software program of claim 59 wherein said resource-constrained device comprises a smart card.
68. (Previously Presented) The software program of claim 59 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).

69. (Previously Presented) An application software program, comprising:
- an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions residing on a computer-readable medium, said instructions comprising operation codes and operands, said program operable to be loaded to and executed by a resource-constrained device, said instructions previously converted from at least one class file, said instructions comprising at least one composite instruction for performing an operation on a current object.
70. (Previously Presented) The software program of claim 69 wherein said resource-constrained device is based on a 16-bit processor architecture.
71. (Previously Presented) The software program of claim 69 wherein said resource-constrained device is based on an 8-bit processor architecture.
72. (Previously Presented) The software program of claim 69 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 64 kilo-bytes.
73. (Previously Presented) The software program of claim 69 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

74. (Previously Presented) The software program of claim 69 wherein said instructions are configured for execution by a virtual machine running on a microprocessor residing on said resource-constrained device.
75. (Previously Presented) The software program of claim 69 wherein said resource-constrained device comprises a smart card.
76. (Previously Presented) The software program of claim 69 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).
- BZ 77. (Previously Presented) A resource-constrained device comprising:  
a memory for storing an application software program comprising an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, said instructions comprising operation codes and operands, said instructions previously converted from at least one class file, said conversion transforming at least one reference to a constant pool to inline data in said instructions; and  
a virtual machine implemented on a microprocessor, said virtual machine configured to execute said sequence of instructions.
78. (Previously Presented) The resource-constrained device of claim 77 wherein one or more of said references to said constant pool are transformed into inline data in operands in one or more of said instructions.


79. (Previously Presented) The resource-constrained device of claim 77 wherein one or more of said references to said constant pool are transformed into inline data in operation codes in one or more of said instructions.
80. (Previously Presented) The resource-constrained device of claim 77 wherein said resource-constrained device is based on a 16-bit processor architecture.
81. (Previously Presented) The resource-constrained device of claim 77 wherein said resource-constrained device is based on an 8-bit processor architecture.
-  82. (Previously Presented) The resource-constrained device of claim 77 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 64 kilo-bytes.
83. (Previously Presented) The resource-constrained device of claim 77 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 4 kilo-bytes.
84. (Previously Presented) The resource-constrained device of claim 77 wherein said resource-constrained device comprises a smart card.

85. (Previously Presented) The resource-constrained device of claim 77 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).
86. (Previously Presented) The resource-constrained device of claim 77 wherein said resource-constrained device comprises a Java Card™ technology-enabled smart card.
87. (Previously Presented) A resource-constrained device comprising:  
a memory for storing an application software program comprising an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, said instructions comprising operation codes and operands, said instructions previously converted from at least one class file, said instructions comprising at least one composite instruction for performing an operation on a current object; and  
a virtual machine implemented on a microprocessor, said virtual machine configured to execute said sequence of instructions.
88. (Previously Presented) The resource-constrained device of claim 87 wherein said resource-constrained device is based on a 16-bit processor architecture.
89. (Previously Presented) The resource-constrained device of claim 87 wherein said resource-constrained device is based on an 8-bit processor architecture.

90. (Previously Presented) The resource-constrained device of claim 87 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 64 kilo-bytes.

91. (Previously Presented) The resource-constrained device of claim 87 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

92. (Previously Presented) The resource-constrained device of claim 87 wherein said resource-constrained device comprises a smart card.

 93. (Previously Presented) The resource-constrained device of claim 87 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).

94. (Previously Presented) The resource-constrained device of claim 87 wherein said resource-constrained device comprises a Java Card™ technology-enabled smart card.

95. (Previously Presented) A method for using an application software program including an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, the method comprising:

receiving said software program in a resource-constrained device having a memory,  
said instructions comprising operation codes and operands, said instructions  
previously converted from at least one class file, said conversion transforming at  
least one reference to a constant pool to inline data in said instructions; and  
executing said sequence of instructions on said resource-constrained device.

96. (Previously Presented) The method of claim 95, further comprising accessing said  
software program over a computer network prior to downloading said program onto  
said resource-constrained device.

B<sup>2</sup> 97. (Previously Presented) The method of claim 95, further comprising storing said  
sequence of instructions on said resource-constrained device.

98. (Previously Presented) The method of claim 95, further comprising accessing said  
software program over the Internet prior to downloading said program onto said  
resource-constrained device.

99. (Previously Presented) The method of claim 95, further comprising transforming  
constant pool indices that appear in the received set of instructions to corresponding  
data values.



100.(Previously Presented) The method of claim 95 wherein one or more of said references to said constant pool are transformed into inline data in operands in one or more of said instructions.

101.(Previously Presented) The method of claim 95 wherein one or more of said references to said constant pool are transformed into inline data in operation codes in one or more of said instructions.

102.(Previously Presented) The method of claim 95 wherein said resource-constrained device is based on a 16-bit processor architecture.

B2  
103.(Previously Presented) The method of claim 95 wherein said resource-constrained device is based on an 8-bit processor architecture.

104.(Previously Presented) The method of claim 95 wherein said memory comprises a random access memory with a capacity of no more than about 64 kilo-bytes.

105.(Previously Presented) The method of claim 95 wherein said memory comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

106.(Previously Presented) The method of claim 95 wherein said instructions are configured for execution by a virtual machine running on a microprocessor residing on said resource-constrained device.

107.(Previously Presented) The method of claim 95 wherein said resource-constrained device comprises a smart card.

108.(Previously Presented) The method of claim 95 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).

BZ  
109.(Previously Presented) A method for using an application software program including an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, the method comprising:  
receiving said software program in a resource-constrained device having a memory,  
said instructions comprising operation codes and operands, said instructions previously converted from at least one class file, said instructions comprising at least one composite instruction for performing an operation on a current object;  
and  
executing said sequence of instructions on said resource-constrained device.

110.(Previously Presented) The method of claim 109, further comprising accessing said software program over a computer network prior to downloading said program onto said resource-constrained device.

111.(Previously Presented) The method of claim 109, further comprising storing said sequence of instructions on said resource-constrained device.

112.(Previously Presented) The method of claim 109, further comprising accessing said software program over the Internet prior to downloading said program onto said resource-constrained device.

113.(Previously Presented) The method of claim 109, further comprising transforming constant pool indices that appear in the received set of instructions to corresponding data values.

114.(Previously Presented) The method of claim 109 wherein one or more of said references to said constant pool are transformed into inline data in operands in one or more of said instructions.

115.(Previously Presented) The method of claim 109 wherein one or more of said references to said constant pool are transformed into inline data in operation codes in one or more of said instructions.

116.(Previously Presented) The method of claim 109 wherein said resource-constrained device is based on a 16-bit processor architecture.

117.(Previously Presented) The method of claim 109 wherein said resource-constrained device is based on an 8-bit processor architecture.

118.(Previously Presented) The method of claim 109 wherein said memory comprises a random access memory with a capacity of no more than about 64 kilo-bytes.

119.(Previously Presented) The method of claim 109 wherein said memory comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

120.(Previously Presented) The method of claim 109 wherein said instructions are configured for execution by a virtual machine running on a microprocessor residing on said resource-constrained device.

b2  
121.(Previously Presented) The method of claim 109 wherein said resource-constrained device comprises a smart card.

122.(Previously Presented) The method of claim 109 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).

123.(Previously Presented) An apparatus for using an application software program including an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, the apparatus comprising:  
means for receiving said software program in a resource-constrained device having a memory, said instructions comprising operation codes and operands, said instructions previously converted from at least one class file, said conversion

transforming at least one reference to a constant pool to inline data in said  
instructions; and  
means for executing said sequence of instructions on said resource-constrained  
device.

124.(Previously Presented) The apparatus of claim 123, further comprising means for  
accessing said software program over a computer network prior to downloading said  
program onto said resource-constrained device.

125.(Previously Presented) The apparatus of claim 123, further comprising means for  
storing said sequence of instructions on said resource-constrained device.

B2  
126.(Previously Presented) The apparatus of claim 123, further comprising means for  
accessing said software program over the Internet prior to downloading said program  
onto said resource-constrained device.

127.(Previously Presented) The apparatus of claim 123, further comprising means for  
transforming constant pool indices that appear in the received set of instructions to  
corresponding data values.

128.(Previously Presented) The apparatus of claim 123 wherein one or more of said  
references to said constant pool are transformed into inline data in operands in one or  
more of said instructions.

129.(Previously Presented) The apparatus of claim 123 wherein one or more of said references to said constant pool are transformed into inline data in operation codes in one or more of said instructions.

130.(Previously Presented) The apparatus of claim 123 wherein said resource-constrained device is based on a 16-bit processor architecture.

131.(Previously Presented) The apparatus of claim 123 wherein said resource-constrained device is based on an 8-bit processor architecture.

32 ✓  
132.(Previously Presented) The apparatus of claim 123 wherein said memory comprises a random access memory with a capacity of no more than about 64 kilo-bytes.

133.(Previously Presented) The apparatus of claim 123 wherein said memory comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

134.(Previously Presented) The apparatus of claim 123 wherein said instructions are configured for execution by a virtual machine running on a microprocessor residing on said resource-constrained device.


135.(Previously Presented) The apparatus of claim 123 wherein said resource-constrained device comprises a smart card.

136.(Previously Presented) The apparatus of claim 123 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).

137.(Previously Presented) An apparatus for using an application software program including an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, the apparatus comprising:

means for receiving said software program in a resource-constrained device having a memory, said instructions comprising operation codes and operands, said instructions previously converted from at least one class file, said instructions comprising at least one composite instruction for performing an operation on a current object; and

means for executing said sequence of instructions on said resource-constrained device.



138.(Previously Presented) The apparatus of claim 137, further comprising means for accessing said software program over a computer network prior to downloading said program onto said resource-constrained device.

139.(Previously Presented) The apparatus of claim 137, further comprising means for storing said sequence of instructions on said resource-constrained device.

140.(Previously Presented) The apparatus of claim 137, further comprising means for accessing said software program over the Internet prior to downloading said program onto said resource-constrained device.

141.(Previously Presented) The apparatus of claim 137, further comprising means for transforming constant pool indices that appear in the received set of instructions to corresponding data values.

142.(Previously Presented) The apparatus of claim 137 wherein one or more of said references to said constant pool are transformed into inline data in operands in one or more of said instructions.

143.(Previously Presented) The apparatus of claim 137 wherein one or more of said references to said constant pool are transformed into inline data in operation codes in one or more of said instructions.

144.(Previously Presented) The apparatus of claim 137 wherein said resource-constrained device is based on a 16-bit processor architecture.

145.(Previously Presented) The apparatus of claim 137 wherein said resource-constrained device is based on an 8-bit processor architecture.



146.(Previously Presented) The apparatus of claim 137 wherein said memory comprises  
a random access memory with a capacity of no more than about 64 kilo-bytes.

147.(Previously Presented) The apparatus of claim 137 wherein said memory comprises  
a random access memory with a capacity of no more than about 4 kilo-bytes.

148.(Previously Presented) The apparatus of claim 137 wherein said instructions are  
configured for execution by a virtual machine running on a microprocessor residing  
on said resource-constrained device.

149.(Previously Presented) The apparatus of claim 137 wherein said resource-  
constrained device comprises a smart card.

150.(Previously Presented) The apparatus of claim 137 wherein said resource-  
constrained device comprises an application-specific integrated circuit (ASIC).

---